NEW **Stack Overflow Jobs** powered by Indeed: A job site that puts thousands of tech jobs at your fingertips (U.S. only). Search jobs

The Definitive C++ Book Guide and List

Asked 15 years, 4 months ago Modified 1 month ago Viewed 3.2m times

4230

This question's answers are a community effort. Edit existing answers to improve this post. It is not currently accepting new answers or interactions.

This question attempts to collect the few pearls among the dozens of bad C++ books that are published every year.

Unlike many other programming languages, which are often picked up on the go from tutorials found on the Internet, few are able to quickly pick up C++ without studying a well-written C++ book. It is way too big and complex for doing this. In fact, it is so big and complex, that ***there are very many very bad C++ books*** out there. And we are not talking about bad style, but things like sporting *glaringly obvious factual errors* and *promoting abysmally bad programming styles*.

Please edit the accepted answer to provide **quality books** and an approximate skill level — *preferably* **after** *discussing your addition in the C++ chat room*. (The regulars might mercilessly undo your work if they disagree with a recommendation.) Add a short blurb/ description about each book that you have personally read/benefited from. Feel free to debate quality, headings, etc. Books that meet the criteria will be added to the list. Books that have reviews by the Association of C and C++ Users (ACCU) have links to the review.

*Note: FAQs and other resources can be found in the C++ tag info and under c++-faq .

c++ c++-faq

Share Follow

edited Jan 18, 2021 at 12:34

community wiki
97 revs, 53 users 23%
sbi

Comments disabled on deleted / locked posts / reviews

1 Answer

Sorted by: Highest score (default)

5973

Beginner

Introductory, no previous programming experience

Book	Author(s)	Description	review
*C++ Primer** * Not to be confused with *C++ Primer Plus* (Stephen Prata), with a significantly less favorable review.	Stanley Lippman, Josée Lajoie, and Barbara E. Moo (**updated for C++11**)	Coming at 1k pages, this is a very thorough introduction into C++ that covers just about everything in the language in a very accessible format and in great detail. The fifth edition (released August 16, 2012) covers C++11.	[Review]
Programming: Principles and Practice Using C++	Bjarne Stroustrup, 2nd Edition - May 25, 2014 (**updated for C++11/C++14**)	An introduction to programming using C++ by the creator of the language. A good read, that assumes no previous programming experience, but is not only for beginners.	

Introductory, with previous programming experience

Book	Author(s)	Description	review
A Tour of C++	Bjarne Stroustrup (**2nd edition for C++17**, **3rd edition for C++20**)	The “tour” is a quick (about 180 pages and 14 chapters) tutorial overview of all of standard C++ (language and standard library, **and using C++11**) at a moderately high level for people who already know C++ or at least are experienced programmers. This book is an extended version of the material that constitutes Chapters 2-5 of The C++ Programming Language, 4th edition.	
Accelerated C++	Andrew Koenig and Barbara Moo, 1st Edition - August 24, 2000	This basically covers the same ground as the *C++ Primer*, but does so in a quarter of its space. This is largely because it does not attempt to be an introduction to *programming*, but an introduction to *C++* for people who've previously programmed in some other language. It has a steeper learning curve, but, for those who can cope with this, it is a very compact introduction to the language. (Historically, it broke new ground by being the first beginner's book to use a modern approach to teaching the language.) Despite this, the C++ it teaches is purely C++98.	[Review]

Best practices

Book	Author(s)	Description	review
		This was written with the aim of being the best second book C+	

Effective C++	Scott Meyers, 3rd Edition - May 22, 2005	+ programmers should read, and it succeeded. Earlier editions were aimed at programmers coming from C, the third edition changes this and targets programmers coming from languages like Java. It presents ~50 easy-to-remember rules of thumb along with their rationale in a very accessible (and enjoyable) style. For C++11 and C++14 the examples and a few issues are outdated and Effective Modern C++ should be preferred.	[Review]
Effective Modern C++	Scott Meyers	This book is aimed at C++ programmers making the transition from C++03 to C++11 and C++14. This book can be treated as a continuation and "correction" of some parts of the previous book - "Effective C++". They don't cover the same things, but keep similar item-based themes.	[Review]
Effective STL	Scott Meyers	This aims to do the same to the part of the standard library coming from the STL what *Effective C++* did to the language as a whole: It presents rules of thumb along with their rationale.	

Intermediate

Book	Author(s)	Description	review
More Effective C++	Scott Meyers	Even more rules of thumb than *Effective C++*. Not as important as the ones in the first book, but still good to know.	
Exceptional C++	Herb Sutter	Presented as a set of puzzles, this has one of the best and thorough discussions of the proper resource management and exception safety in C++ through Resource Acquisition is Initialization (RAII) in addition to in-depth coverage of a variety of other topics including the pimpl idiom, name lookup, good class design, and the C++ memory model.	[Review]
More Exceptional C++	Herb Sutter	Covers additional exception safety topics not covered in *Exceptional C++*, in addition to discussion of effective object-oriented programming in C++ and correct use of the STL.	[Review]
Exceptional C++ Style	Herb Sutter	Discusses generic programming, optimization, and resource management; this book also has an excellent exposition of how to write modular code in C++ by using non-member functions and the single responsibility principle.	[Review]
C++ Coding Standards	Herb Sutter and Andrei	“Coding standards” here doesn't mean “how many spaces should I indent my code?” This book contains 101 best practices, idioms, and common pitfalls that can	[Review]

	Alexandrescu	help you to write correct, understandable, and efficient C++ code.	
C++ Templates: The Complete Guide	David Vandevoorde and Nicolai M. Josuttis	This is *the* book about templates as they existed before C++11. It covers everything from the very basics to some of the most advanced template metaprogramming and explains every detail of how templates work (both conceptually and at how they are implemented) and discusses many common pitfalls. Has excellent summaries of the One Definition Rule (ODR) and overload resolution in the appendices. A second edition covering C++11, C++14 and C++17 has been already published.	[Review]
C++ 17 - The Complete Guide	Nicolai M. Josuttis	This book describes all the new features introduced in the C++17 Standard covering everything from the simple ones like 'Inline Variables', 'constexpr if' all the way up to 'Polymorphic Memory Resources' and 'New and Delete with over aligned Data'.	[Review]
C++ 20 - The Complete Guide	Nicolai M. Josuttis	This book presents all the new language and library features of C++20. It covers the motivation and context of each new feature with examples and background information. The focus is on how these features impact day-to-day programming, what it means to combine them, and how to benefit from C++20 in practice. (Note that this book was **published step-by-step**, and the first edition is now complete.)	
C++ in Action	Bartosz Milewski	This book explains C++ and its features by building an application from the ground up.	[Review]
Functional Programming in C++	Ivan Čukić	This book introduces functional programming techniques to modern C++ (C++11 and later). A very nice read for those who want to apply functional programming paradigms to C++.	

Advanced

Book	Author(s)	Description	review
Modern C++ Design	Andrei Alexandrescu	A groundbreaking book on advanced generic programming techniques. Introduces policy-based design, type lists, and fundamental generic programming idioms then explains how many useful design patterns (including small object allocators, functors, factories, visitors, and multi-methods) can be implemented efficiently, modularly, and cleanly using generic programming.	[Review]

C++ Template Metaprogramming	David Abrahams and Aleksey Gurtovoy		
C++ Concurrency In Action	Anthony Williams	A book covering C++11 concurrency support including the thread library, the atomics library, the C++ memory model, locks and mutexes, as well as issues of designing and debugging multithreaded applications. A second edition covering C++14 and C++17 has already been published.	[Review]
Advanced C++ Metaprogramming	Davide Di Gennaro	A pre-C++11 manual of TMP techniques, focused more on practice than theory. There are a ton of snippets in this book, some of which are made obsolete by type traits, but the techniques, are nonetheless useful to know. If you can put up with the quirky formatting/editing, it is easier to read than Alexandrescu, and arguably, more rewarding. For more experienced developers, there is a good chance that you may pick up something about a dark corner of C++ (a quirk) that usually only comes about through extensive experience.	
Large Scale C++ volume I, Process and architecture (2020)	John Lakos	Part one of a three-part series extending the older book 'Large Scale C++ Design'. Lakos explains battle-tested techniques to manage very big C++ software projects. If you work in a big C++ software project this is a great read, detailing the relationship between physical and logical structure, strategies for components, and their reuse.	[Review]

Reference Style - All Levels

Book	Author(s)	Description	review
The C++ Programming Language	Bjarne Stroustrup (**updated for C++11**)	The classic introduction to C++ by its creator. Written to parallel the classic K&R, this indeed reads very much like it and covers just about everything from the core language to the standard library, to programming paradigms to the language's philosophy.	[Review] Note: All releases of the C++ standard are tracked in the question "*Where do I find the current C or C++ standard documents?*".
C++ Standard Library Tutorial and Reference	Nicolai Josuttis (**updated for C++11**)	*The* introduction and reference for the C++ Standard Library. The second edition (released on April 9, 2012) covers C++11.	[Review]

	C++11)		
The C++ IO Streams and Locales	Angelika Langer and Klaus Kreft	There's very little to say about this book except that if you want to know anything about streams and locales, then this is the one place to find definitive answers.	[Review]

C++11/14/17/… References:

- Working Draft, Standard for Programming Language C++ generated from LaTeX sources published on GitHub.
- C++ Standard Papers, latest standard working draft: ISO working draft
- *The C++11/14/17 Standard (INCITS/ISO/IEC 14882:2011/2014/2017)* This, of course, is the final arbiter of all that is or isn't C++. Be aware, however, that it is intended purely as a reference for *experienced* users willing to devote considerable time and effort to its understanding. The C++17 standard is released in electronic form for 198 Swiss Francs.
- The C++17 standard is available, but seemingly not in an economical form – directly from the ISO it costs 198 Swiss Francs (about $200 US). For most people, the final draft before standardization is more than adequate (and free). Many will prefer an even newer draft, documenting new features that are likely to be included in C++20.
- C++20 draft is available on GitHub as some older too.
- *Overview of the New C++ (C++11/14) (PDF only)* (Scott Meyers) (**updated for C++14**) These are the presentation materials (slides and some lecture notes) of a three-day training course offered by Scott Meyers, who's a highly respected author on C++. Even though the list of items is short, the quality is high.
- The *C++ Core Guidelines (C++11/14/17/...)* (edited by Bjarne Stroustrup and Herb Sutter) is an evolving online document consisting of a set of guidelines for using modern C++ well. The guidelines are focused on relatively higher-level issues, such as interfaces, resource management, memory management, and concurrency affecting application architecture and library design. The project was announced at CppCon'15 by Bjarne Stroustrup and others and welcomes contributions from the community. Most guidelines are supplemented with a rationale and examples as well as discussions of possible tool support. Many rules are designed specifically to be automatically checkable by static analysis tools.
- The *C++ Super-FAQ* (Marshall Cline, Bjarne Stroustrup, and others) is an effort by the Standard C++ Foundation to unify the C++ FAQs previously maintained individually by Marshall Cline and Bjarne Stroustrup and also incorporating new contributions. The items mostly address issues at an intermediate level and are often written with a humorous tone. Not all items might be fully up to date with the latest edition of the C++ standard yet.
- *cppreference.com (C++03/11/14/17/...)* (initiated by Nate Kohl) is a wiki that summarizes

the basic core-language features and has extensive documentation of the C++ standard library. The documentation is very precise but is easier to read than the official standard document and provides better navigation due to its wiki nature. The project documents all versions of the C++ standard and the site allows filtering the display for a specific version. The project was presented by Nate Kohl at CppCon'14.

Classics / Older

Note: Some information contained within these books may not be up-to-date or no longer considered best practice.

- *The Design and Evolution of C++* (Bjarne Stroustrup) If you want to know *why* the language is the way it is, this book is where you find answers. This covers everything *before the standardization* of C++.
- *Ruminations on C++* - (Andrew Koenig and Barbara Moo) [Review]
- *Advanced C++ Programming Styles and Idioms* (James Coplien) A predecessor of the pattern movement, it describes many C++-specific “idioms”. It's certainly a very good book and might still be worth a read if you can spare the time, but quite old and not up-to-date with current C++.
- *Large Scale C++ Software Design* (John Lakos) Lakos explains techniques to manage very big C++ software projects. Certainly, a good read, if it only was up to date. It was written long before C++ 98 and misses on many features (e.g. namespaces) important for large-scale projects. If you need to work on a big C++ software project, you might want to read it, although you need to take more than a grain of salt with it. Not to be confused with the extended and later book series Large Scale C++ volume I-III.
- *Inside the C++ Object Model* (Stanley Lippman) If you want to know how virtual member functions are commonly implemented and how base objects are commonly laid out in memory in a multi-inheritance scenario, and how all this affects performance, this is where you will find thorough discussions of such topics.
- *The Annotated C++ Reference Manual* (Bjarne Stroustrup, Margaret A. Ellis) This book is quite outdated in the fact that it explores the 1989 C++ 2.0 version - Templates, exceptions, namespaces, and new casts were not yet introduced. Saying that however, this book goes through the entire C++ standard of the time explaining the rationale, the possible implementations, and features of the language. This is not a book to learn programming principles and patterns on C++, but to understand every aspect of the C++ language.
- *Thinking in C++* (Bruce Eckel, 2nd Edition, 2000). Two volumes; is a tutorial-style *free* set of intro-level books. Downloads: vol 1, vol 2. Unfortunately, they're marred by a number of trivial errors (e.g. maintaining that temporaries are automatic `const`), with no official errata list. A partial 3rd party errata list is available at http://www.computerscienceslab.com/Eckel.htm, but it is apparently not maintained

www.computersciencelab.com/EckelTalk.htm, but it is apparently not maintained.

- *Scientific and Engineering C++: An Introduction to Advanced Techniques and Examples* (John Barton and Lee Nackman) It is a comprehensive and very detailed book that tried to explain and make use of all the features available in C++, in the context of numerical methods. It introduced at the time several new techniques, such as the Curiously Recurring Template Pattern (CRTP, also called Barton-Nackman trick). It pioneered several techniques such as dimensional analysis and automatic differentiation. It came with a lot of compilable and useful code, ranging from an expression parser to a Lapack wrapper. The code is still available online. Unfortunately, the books have become somewhat outdated in style and C++ features, however, it was an incredible tour-de-force at the time (1994, pre-STL). The chapters on dynamics inheritance are a bit complicated to understand and not very useful. An updated version of this classic book that includes move semantics and the lessons learned from the STL would be very nice.

Share Improve this answer Follow

edited Apr 3 at 1:57

community wiki
118 revs, 90 users 14%
George Stocker

543 @G Rassovsky: All books which promise to teach X in Y hours. For example Learn C++ in 24 hours. I believe all such books are better avoided. – akhil_mittal Dec 29, 2014 at 4:47

161 I hate to step on anybody's shoes, but I do not recommend Bruce Eckel's "Thinking in C++" even though I respect the author for publishing his materials online for free. The book's perspective suggests relatively poor or ineffective use of C++ and "object oriented" programming, akin to poor application of the GoF Design Patterns. I found it an interesting introductory book to programming in general, but as someone becomes more familiarized with programming and (especially) computer science as a whole, I find books which think purely in "classic" OOP terms detrimental to education. – user1630889 Jan 16, 2015 at 5:55

44 @G.Rassovsky on the accu.org website, there's a book reviews section with ratings. You can search for the C++ ones. Many of them are rated "not recommended". – Zaphod Beeblebrox Jan 26, 2016 at 21:58

21 This one is a very important comment. Moderators please do not delete. I have an anti recommendation here: "Let Us C++ by Yashavant Kanetkar". This is a complete trash. I request all beginners/programmers do not read this book. Reading this book is like it will teach you `2+2 = 4` in the chapters and then in the exercise it will ask you to calculate the area of universe. Highly demotivating. – haccks Apr 3, 2022 at 16:54

12 Could C++ Move Semantics - The Complete Guide by Nicolai Josuttis be added to the list? – Myrddin Krustowski Jun 14, 2022 at 8:22